ΕΠΛ 361 ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ ΕΡΓΑΣΤΗΡΙΟ 1

Tools

UML

- Η Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language) είναι μια γραφική γλώσσα για την οπτική παράσταση, τη διαμόρφωση προδιαγραφών και την τεκμηρίωση συστημάτων που βασίζονται σε λογισμικό. Η UML στοχεύει στο σχεδιασμό αντικειμενοστρεφών συστημάτων. Το σχέδιο είναι μια απλοποιημένη παράσταση της πραγματικότητας.
- Σχεδιάζουμε για να μπορέσουμε να καταλάβουμε το σύστημα που αναπτύσσουμε. Έτσι δημιουργώντας ένα σχέδιο επιτυγχάνουμε τέσσερεις στόχους:
  - παριστάνουμε οπτικά το σύστημα που έχουμε ή θέλουμε να κατασκευάσουμε,
  - προσδιορίζουμε τη δομή και τη συμπεριφορά του συστήματος,
  - δημιουργούμε ένα πρότυπο για να βασίσουμε την κατασκευή του συστήματος,
  - τεκμηριώνουμε τις αποφάσεις που λάβαμε.

UML

- Η UML είναι μία γλώσσα μοντελοποίησης (σύνολο από διαγράμματα). (**Δεν είναι** γλώσσα προγραμματισμού)
- Η UML αποτελείται από ένα σύνολο κανόνων και συμβόλων για την περιγραφή και σχεδίαση ενός συστήματος λογισμικού.

- Τα σύμβολα παρέχουν ένα σύνολο γραφικών στοιχείων για την μοντελοποίηση κάθε μέρους του συστήματος.

- UML Tools
  - Υπάρχουν μια σειρά από εργαλεία UML που είναι διαθέσιμα τόσο εμπορικά όσο και ανοικτού κώδικα, για να σας βοηθήσουν να τεκμηριώνετε τα σχέδιά σας. Standalone tools, plug-ins and UML editors are available for most IDEs.
  - ΕΠΛ361 -> Modelio, ArgoUML

UML

- Η UML 2 αποτελείται από ένα σύνολο από διαγράμματα
- **STRUCTURAL DIAGRAMS 1/2**
  - **Διαγράμματα Κλάσης (Class Diagrams) -** περιγράφουν τη στατική δομή των κλάσεων στο σύστημά σας και απεικονίζουν τα χαρακτηριστικά, τις λειτουργίες και τις σχέσεις μεταξύ των κλάσεων.
  - **Διαγράμματα Αντικειμένων (Object Diagrams) -** παρέχουν πληροφορίες σχετικά με τις σχέσεις μεταξύ των instances των κλάσεων σε μια συγκεκριμένη χρονική στιγμή.
  - **Διαγράμματα Συνιστώσας (Component Diagrams) -** χρησιμοποιείται για να δείξει πως συνδέονται μαζί τα components του συστήματος σε ένα υψηλότερο επίπεδο αφαιρετικότητας από ότι τα διαγράμματα κλάσεων. Ένα component θα μπορούσε να μοντελοποιηθεί από μία ή περισσότερες κλάσεις.

UML

- **STRUCTURAL DIAGRAMS 2/2**
  - **Composite Structure Diagrams -** δείχνουν την εσωτερική δομή μιας κλάσης.
  - **Deployment Diagrams -** μοντελοποιούν την αρχιτεκτονική του συστήματος σε ένα πραγματικό περιβάλλον. Δείχνουν πώς αναπτύσσονται οι οντότητες λογισμικού σε φυσικούς κόμβους υλικού και συσκευών.
  - **Package Diagrams –** δείχνουν την οργάνωση των packages

UML

- **BEHAVIORAL DIAGRAMS**
  - **Διαγράμματα Περιπτώσεων Χρήσης (Use Case Diagrams)**
    - Παρουσιάζουν τους actors και τις περιπτώσεις χρήσης ενός συστήματος
    - Οι περιπτώσεις χρήσης αναπαριστούν την λειτουργικότητα του συστήματος
  - **Διαγράμματα Δραστηριότητας (Activity Diagrams)**
    - Ένα διάγραμμα δραστηριοτήτων περιγράφει την ροή των εργασιών μέσα στο σύστημα
    - Ένα διάγραμμα δραστηριοτήτων βασικά περιέχει : Δραστηριότητες (activities), Ενέργειες (actions) και Μεταβάσεις (transitions)
    - Τα διαγράμματα δραστηριοτήτων χρησιμοποιούνται για την περιγραφή :
      - μιας περίπτωσης χρήσης
      - ροών εργασιών (workflows)
      - διεργασιών (processes)
      - Επιχειρηματικών διαδικασιών

UML

- **Διαγράμματα Κατάστασης (State Machine Diagrams)**
- Ένα διάγραμμα καταστάσεων περιγράφει μια μηχανή καταστάσεων δίνοντας έμφαση στην ροή του ελέγχου από κατάσταση σε κατάσταση
- Μια μηχανή καταστάσεων προσδιορίζει
  - Τις καταστάσεις (states) που μπορεί να βρεθεί ένα αντικείμενο
  - Τα γεγονότα (events) στα οποία αντιδρά ένα αντικείμενο
  - Την απόκριση (response) του αντικειμένου στα γεγονότα
- Ένα διάγραμμα καταστάσεων περιέχει
  - Καταστάσεις (states) και
  - Μεταβάσεις (transitions)
- Ένα διάγραμμα καταστάσεων χρησιμοποιείται για την περιγραφή ενεργών (reactive) αντικειμένων

UML

- **INTERACTION DIAGRAMS**
  - **Διαγράμματα Ακολουθίας (Sequence Diagrams)**
    - παρουσιάζουν την αλληλεπίδραση των αντικειμένων μέσω της ανταλλαγής μηνυμάτων
      - Δίνουν έμφαση στη χρονική αλληλουχία των μηνυμάτων
    - Τα διαγράμματα ακολουθίας χρησιμοποιούνται για να περιγράψουν τον κύκλο ζωής των αντικειμένων
    - describe how entities interact, including what messages are used for the interactions
    - messages are described in the order of execution.
    - the most used diagrams along with class and use case,
  - **Communication Diagrams**
    - similar to sequence diagrams, except that they are defined in free form instead of lifelines.
  - **Interaction Overview Diagrams**
    - a form of activity diagram where each node is a link to another type of interaction diagram.

MS Project

- **Microsoft Project**
  - a project management software program,
  - developed and sold by Microsoft,
  - designed to assist a project manager in:
    - developing a plan,
    - assigning resources to tasks,
    - tracking progress,
    - managing the budget, and
    - analyzing workloads.

Dreamweaver

- What is Adobe Dreamweaver?
  - A web development tool that lets you create dynamic websites
- What is a website?
  - A group of related web pages that are linked together and share a common interface and design

Dreamweaver

- What does Dreamweaver CS5 offer?
  - Design tools that can create dynamic and interactive web pages without writing HTML code
  - Organizational tools
  - Site management tools

Plan and Set Up a Website

Phases of a website development project

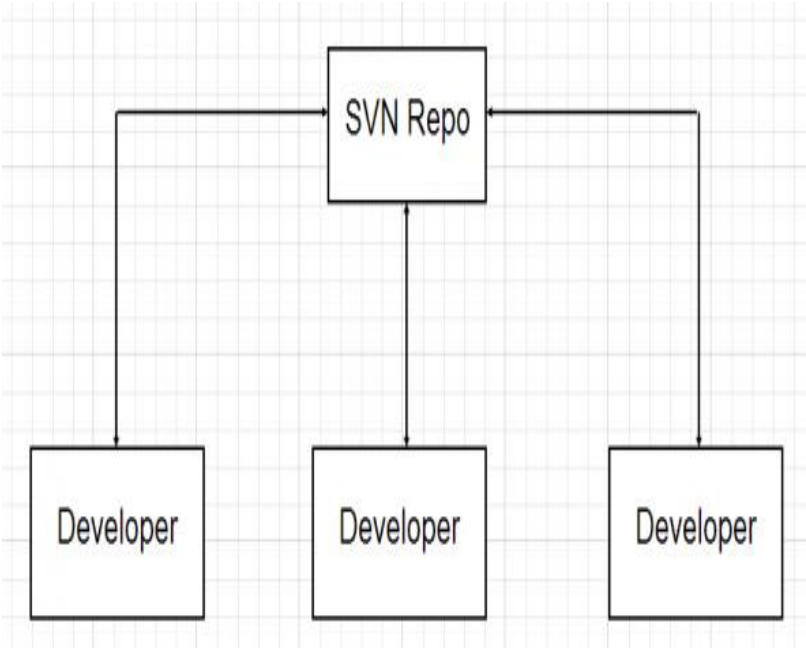
Διαχείριση Εκδόσεων

- Για τη διαχείριση των εγγράφων και του πηγαίου κώδικα κάθε ομάδας:
  - Apache Subversion (SVN)
  - GitHub

Βασικές Έννοιες του SVN

- The Repository
  - Το SVN χρησιμοποιεί μια κεντρική βάση δεδομένων που περιέχει όλα τα ελεγχόμενα αρχεία σας, μαζί με ολόκληρο το ιστορικό των εκδόσεων τους
  - Αυτό το repository συνήθως υπάρχει σε ένα file server που τρέχει το server πρόγραμμα του SVN, το οποίο προσφέρει περιεχόμενο στους διάφορους clients του SVN (όπως στο TortoiseSVN) όταν ζητείται
- Working Copy
  - Κάθε developer έχει το δικό του working copy τοπικά στον Η/Υ του
  - Μπορείτε να κατεβάσετε την τρέχουσα έκδοση κάποιου αρχείου από το repository, να την επεξεργαστείτε τοπικά χωρίς να επηρεάσετε τους υπόλοιπους developer, και εφόσον είσαστε ικανοποιημένοι με τις αλλαγές που κάνατε να τις δεσμεύσετε πίσω στο repository
  - Ένα working copy του SVN δεν περιέχει όλο το ιστορικό του έργου, αλλά κρατά αντίγραφο των αρχείων καθώς υπάρχουν στο repository πριν να ξεκινήσετε να κάνετε τις αλλαγές σας, κάτι το οποίο σημαίνει ότι είναι εύκολο να ελέγξετε ακριβώς ποιες αλλαγές έχετε κάνει

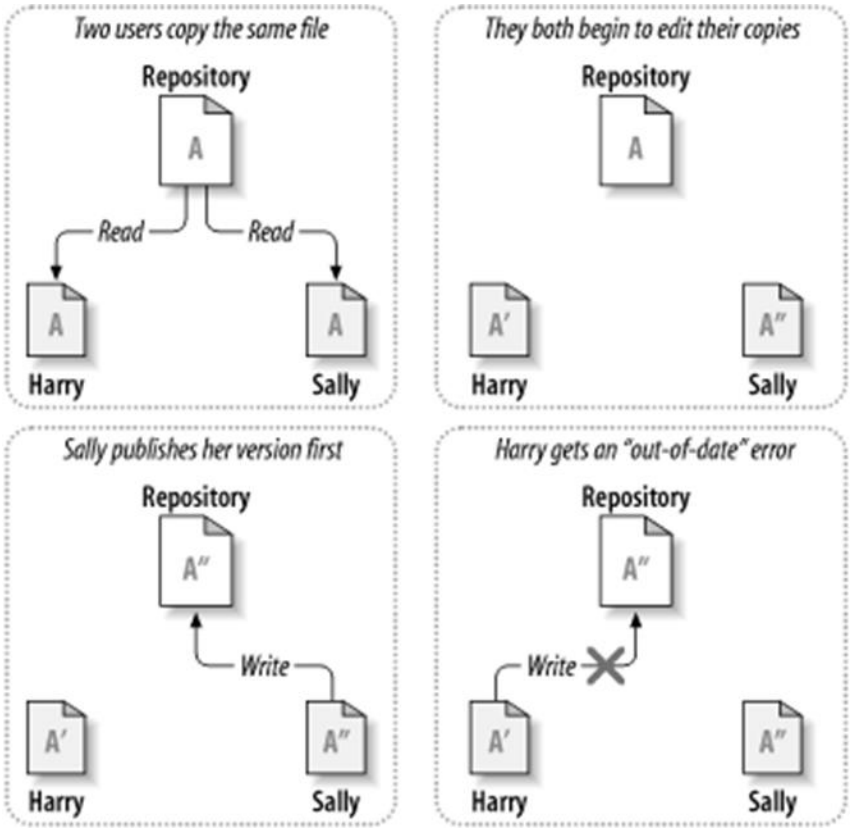
svn



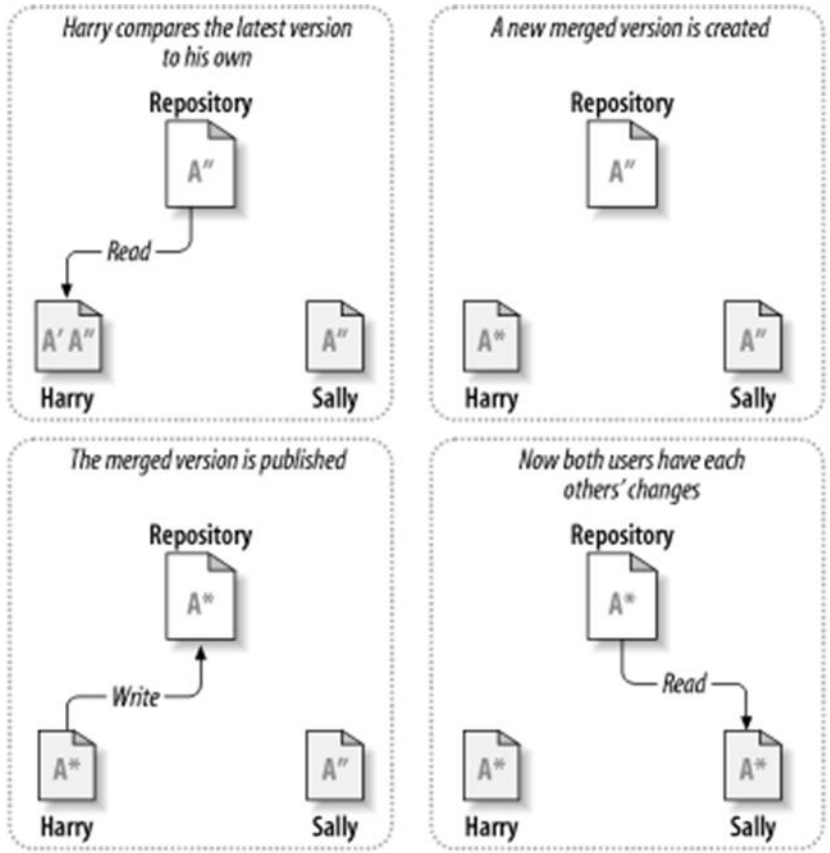
- It works by having a central server for your repository,
- this repository is split into 3 key areas:
  - Trunk (where your stable code will live)
  - Branches (used when you want to create a new feature, so you branch the code from the trunk)
  - Tags (a way of marking your code as a certain points in time)
    - The only difference between a tag and a branch is that tags should never be used for developing, they are there for an easy way of reverting your code back.

Μέθοδος File Sharing στο SVN

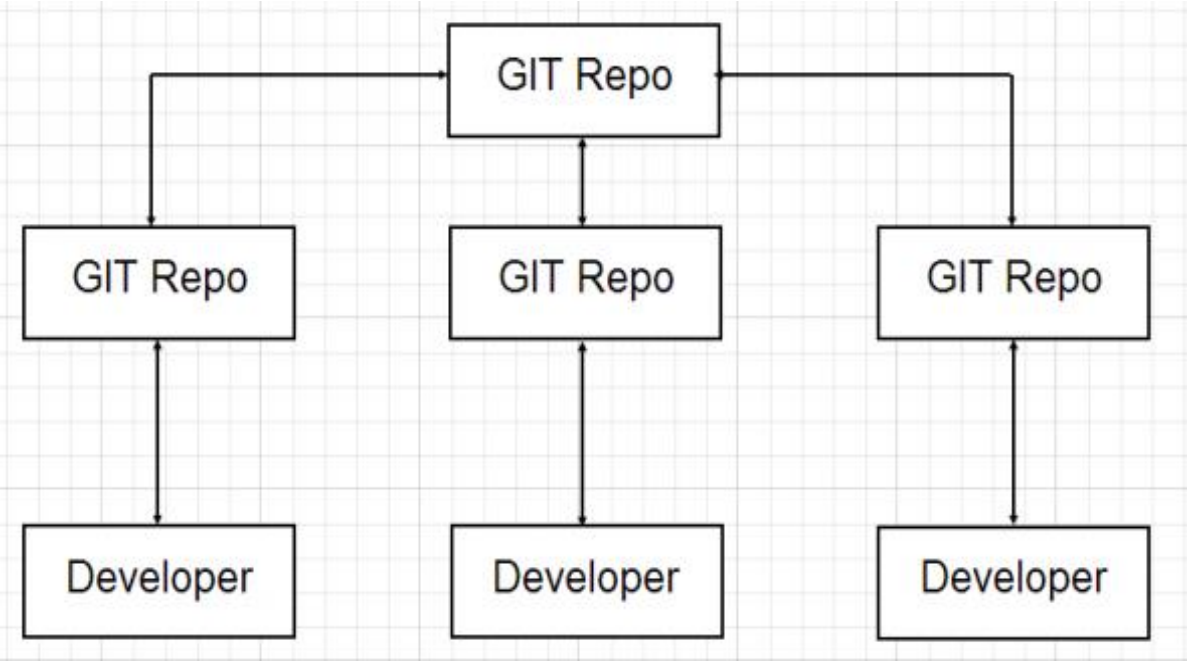
Copy-Modify-Merge (1)



Copy-Modify-Merge (2)



Git

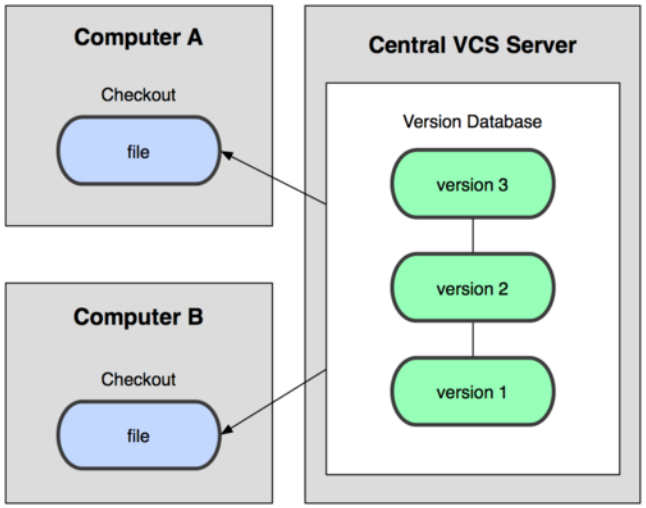

GIT Repo
GIT Repo
GIT Repo
GIT Repo
Developer
Developer
Developer

Git History

- Came out of Linux development community
- Linus Torvalds, 2005
- Initial goals:
  - Speed
  - Support for non-linear development (thousands of parallel branches)
  - Fully distributed
  - Able to handle large projects like Linux efficiently

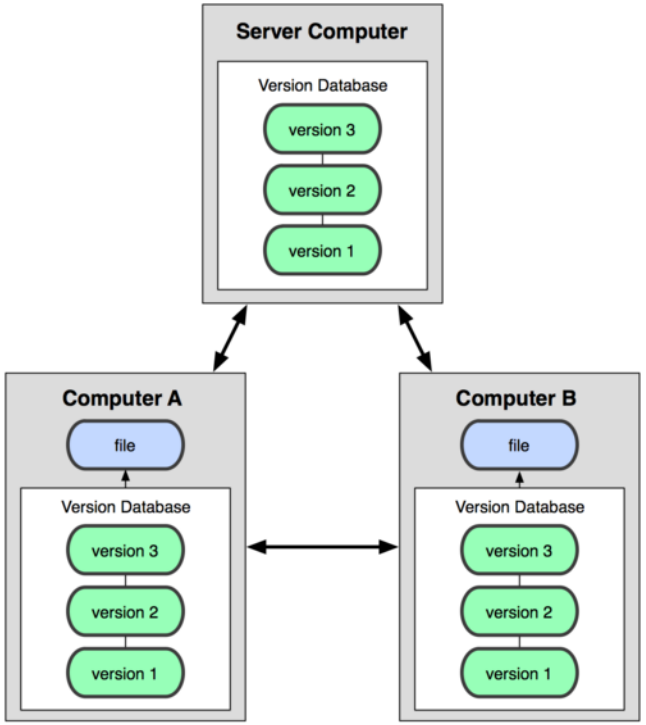
Git uses a distributed model

Centralized Model



(CVS, Subversion, Perforce)

Distributed Model



(Git, Mercurial)

Result: Many operations are local

Basic Workflow

Basic Git workflow:

1. **Modify** files in your working directory.
2. **Stage** files, adding snapshots of them to your staging area.
3. Do a **commit**, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.

- Notes:
  - If a particular version of a file is in the **git directory**, it's considered **committed**.
  - If it's modified but has been added to the **staging area**, it is **staged**.
  - If it was **changed** since it was checked out but has not been staged, it is **modified**.

Aside: So what is github?

- Το GitHub είναι μια Git web-based υπηρεσία φιλοξενίας αποθετηρίων που προσφέρει όλες τις λειτουργίες του Git, καθώς και πολλά δικά της χαρακτηριστικά.
- Σε αντίθεση με το Git που είναι ένα εργαλείο γραμμής εντολών, το Github παρέχει ένα γραφικό περιβάλλον επιφάνειας εργασίας, ένα web-based, καθώς και έκδοση για κινητά.
- Παρέχει, επίσης, έλεγχο πρόσβασης και πολλές δυνατότητες συνεργασίας, όπως wikis, task management, και εργαλεία παρακολούθησης σφαλμάτων και feature requests για κάθε έργο.
- Many open source projects use it, such as the Linux kernel.
- You can get free space for open source projects or you can pay for private projects.

Question: Do I have to use github to use Git?

Answer: No!

- you can use Git completely locally for your own purposes

SVN vs. Git

- SVN:
  - κεντρικό repository. Το repository του server είναι το κύριο με όλη την ιστορία των αλλαγών.
  - Οι χρήστες κάνουν check out τοπικά αντίγραφα της τρέχουσας έκδοσης
- Git:
  - κατανεμημένο repository. Κάθε check out είναι ένα πλήρες repository με όλη την ιστορία του κώδικα.
  - Προσφέρει ταχύτητα και redundancy(πλεονασμό).
  - Οι διαδικασίες συγχώνευσης(merging) και διακλάδωσης (branching) είναι πολύ πιο συχνές